ARMY RESEARCH LABORATORY

The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files

by Binh Q. Nguyen

ARL-TR-4600 **September 2008**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-4600 **September 2008**

The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files

Binh Q. Nguyen
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.			
1. REPORT DATE *(DD-MM-YYYY)* September 2008	2. REPORT TYPE Final		3. DATES COVERED (From - To) Fiscal Year 2008
4. TITLE AND SUBTITLE The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files			5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Binh Q. Nguyen			5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-NT 2800 Powder Mill Road Adelphi, MD 20783-1197			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-4600
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approve for public release; distribution unlimited.			
13. SUPPLEMENTARY NOTES			
14. ABSTRACT The U.S. Army Research Laboratory (ARL) designed and developed a tool called the ARL Real-time Application Representative (RaprEdt) tool to support the emulation of communication scenarios by providing a method for rapidly creating critical files that are required for using the RAPR Version 1.0, a product of the U.S. Naval Research Laboratory (NRL). This report documents the functional behavior of the ARL RaprEdt tool and describes its graphical user interfaces (GUIs).			
15. SUBJECT TERMS RAPR, editor, GUI			
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 20
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	19a. NAME OF RESPONSIBLE PERSON Binh Q. Nguyen 19b. TELEPHONE NUMBER (*Include area code*) (301) 394-1781

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Summary

The Real-time Application Representative (RAPR) Version 1.0 is a product of the U.S. Naval Research Laboratory (NRL). It is designed mainly to generate communication traffic. Its behavior is defined by the textual specifications stored in three different types of computer files: (1) a dictionary file, (2) a logic-table file, and (3) script files. The three file types are collectively called RAPR files. The files are used to specify a communication scenario in a laboratory environment. They define the role and the behavior of each participating computer in a network test bed.

Experimenting with various communication scenarios in a test bed requires distinguishable RAPR files for each scenario. Each scenario involves different number of communicating entities. When the number of communicating entities increases, the manual creation of the files would be a tedious, error-prone, and time-consuming process. Circumventing the manual process necessitate the development of an automated tool called “the U.S. Army Research Laboratory (ARL) RaprFileEDT tool” capable of generating RAPR files more rapidly.

Learning how to use the ARL RaprEdt tool requires very little time because it is equipped with graphical user interfaces with which the user is very familiar. Using the tool, a communication engineer can focus on the development of a communication scenario without being concerned with the mechanical structure of the files.

INTENTIONALLY LEFT BLANK.

1. Background

The U.S. Army Research Laboratory (ARL) uses the Real-time Application Representative[1] (RAPR) Version 1.0 to generate and response to communication traffic and events in the wireless emulation laboratory (WEL), which has been designed and constructed to emulate a dynamic movement of a mobile ad hoc network (MANET). It is a high-performance test bed consisting of a gigabit network connecting 48 physical computers. Each computer is capable of hosting a sub-network of virtual machines (VMs) loaded with a version of the Linux® operating system to host advanced MANET routing protocols, networked applications, information-assurance (IA) products, and performance measurement and visualization tools. The WEL test bed is also a showcase for ARL innovation and successes.

The U.S. Naval Research Laboratory (NRL) designs and develops the RAPR program for experimentation purposes. The RAPR program is used to generate communication traffic and to measure and capture its performance metrics for subsequent analysis. The behavior of a communication scenario is defined by the contents of the files that are associated with a running RAPR program and that are designed specifically for a particular scenario. The files are collectively called RAPR files consisting of dictionary, logic table, and script files.

Experimenting with various communication scenarios in the ARL WEL test bed requires a set of RAPR files for each scenario. Each scenario involves different number of communicating entities. Some act as servers and some act as clients. The RAPR files are used to define the roles and their related behavior of the participating MANET nodes. The number of mobile nodes that can be emulated in the WEL test bed is increasing rapidly. The WEL test bed now can accommodate the emulation of a 100-node MANET, and it is being improved to handle a 1000-node MANET.

Because the manual creation of RAPR files for large networks consisting of hundreds of nodes would be a tedious and time-consuming process, ARL requires that the process be augmented with an automated tool capable of facilitating and expediting the process. In fiscal year 2008 (FY08), ARL conducted a requirement analysis that resulted in a set of performance specifications for an automated tool[2] that would enable the creation of RAPR files more rapidly by freeing its users from being concerned with the required detailed structure of the files. The specifications were then used to develop the ARL RaprEdt tool, also known as the ARL RaprFileEDT, which is reported in this report.

[1] Networks and Communications Systems Branch, "RAPR - The Real-Time Application Representative, The U.S. Naval Research Laboratory, Code 5520, 4555 Overlook Ave., SW, Washington, DC 20375-5337. http://cs.itd.nrl.navy.mil/work/rapr (accessed 05 November 2007).

[2] Hardy, R.; Nguyen, B. ,*Performance Requirements of Tools and Methods for Specifying Network Communication Scenarios Using the Real-Time Application Representative Version 1.0*; ARL-TR-4614; U.S. Army Research Laboratory: Adelphi, MD 20783, September 2008.

Subsequent sections of this report describe the ARL RaprEdt tool and include the screenshots illustrating the features that are built into the tool. The final section concludes the report with a plan for integrating the tool with other ARL-developed tools to support the development, test, and evaluation of MANET technologies and applications.

2. The ARL RaprEdt Tool

The RaprEdt tool is an editing tool equipped with graphical user interface (GUI) features. The tool was created using the Python programming language and its built-in Tkinter module[3]. The tool is an integrated tool comprising three editing tools: (1) the dictionary editor, (2) the logic-table editor, and (3) the script editor. The three editing tools share the main GUI, which is used for selecting a file type, saving and viewing a file, and observing the results of an editing session.

Figure 1 shows the initial appearance of the tool and its usage instructions, which are displayed whenever a menu item in the **Help** menu is pulled down and selected. This is the first step a new user would do before using the tool to create a RAPR file.

The **File** menu of the tool, being placed at the upper-left corner of the screen, has a list of menu items that deal with a RAPR file. This section describes the behavior of three commonly performed actions: creating a new file or editing and viewing an existing one. Other menu items, such as the **Save** operations, are not described in this report because their functional behavior is self-explanatory and similar to many other software applications.

[3] The Python Programming Language. http://www.python.org (accessed 08 July 2008).

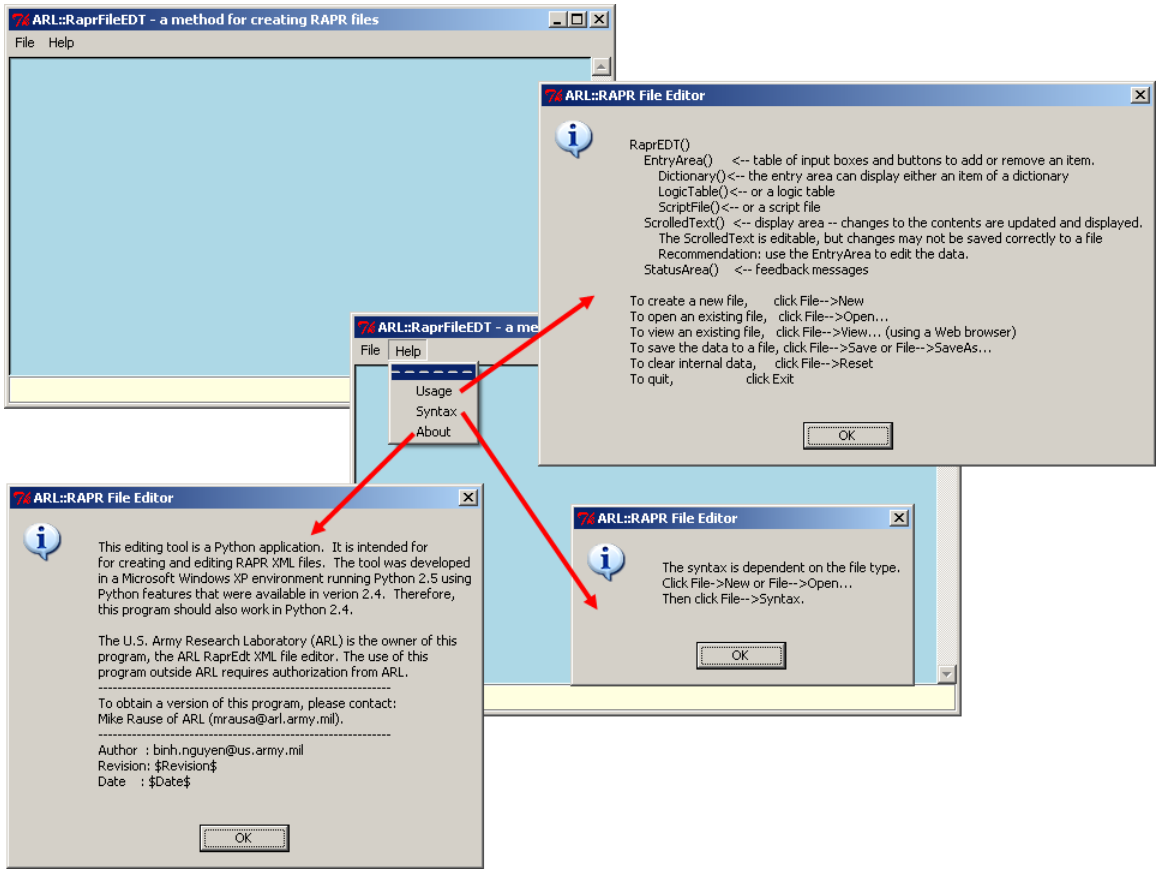

Figure 1. Appearance and usage instructions of the RaprEdt tool.

Figure 2 shows the screenshots when a **File** menu item is selected and the ensuing actions that the user needs to perform. To create a new RAPR file, the user needs to select the menu item labeled **New** and specifies a file type. When the **New** menu item is selected, the tool opens a dialog window displaying three radio buttons indicating the three different file types: dictionary, logic table, and script. The user then selects a file type by clicking an appropriate radio button and the **OK** button. The first two file types contain the text being marked up using the extensible markup language (XML), and the last type is intended for storing a sequence of RAPR scripts, consisting of timed events and commands.

Selecting the menu item **Open** or **View** opens a file-dialog window showing a list of existing XML files from which the user can choose one. If the **Open** operation is selected, then the contents of the selected XML file are loaded into the tool, which automatically determines its file type, i.e., whether it is a dictionary or a logic-table file. If the **View** operation is selected, then the contents of the selected XML file are loaded and displayed in the default Web browser of the host system on which the tool runs. Figure 2 shows the Microsoft Internet Explorer displaying the contents of a RAPR dictionary file.

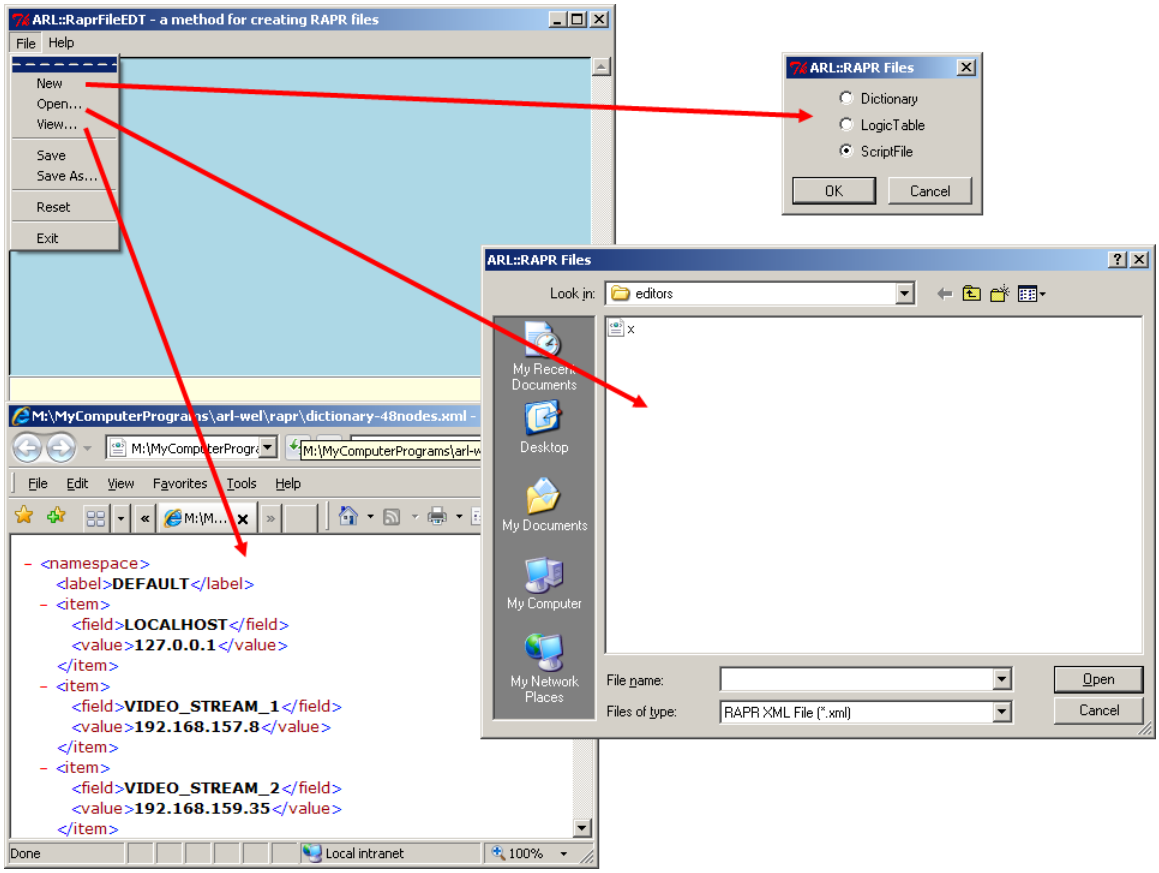

Figure 2. Selecting a type of RAPR file.

2.1 The RAPR Dictionary Editor

A RAPR dictionary contains a set of unique keys or names that are used to look up their associated values. A RAPR dictionary is used to translate name-value pairs used in RAPR files. Each dictionary has one or more **namespace** fields. Each namespace has a **label** field and one or more **item** fields. Each field is given a unique name and assigned one or more values.

When a RAPR dictionary is being edited, the ARL RaprEdt tool displays a set of editing options that are specifically designed for editing a RAPR dictionary. Figure 3 displays the screenshot of the ARL RaprEdt tool when it deals with a RAPR dictionary. The rules for creating a dictionary and the XML tags are shown to its users for informational purposes only, i.e., the user does not need to deal with the XML tags. Figure 3 also shows two examples. The first example illustrates the entry of single name-value pair into a dictionary, and the second the entry of multiple name-value pairs, which is specified in a single line within the field entry box. Note that the entries in the dictionary are not alphabetically sorted.

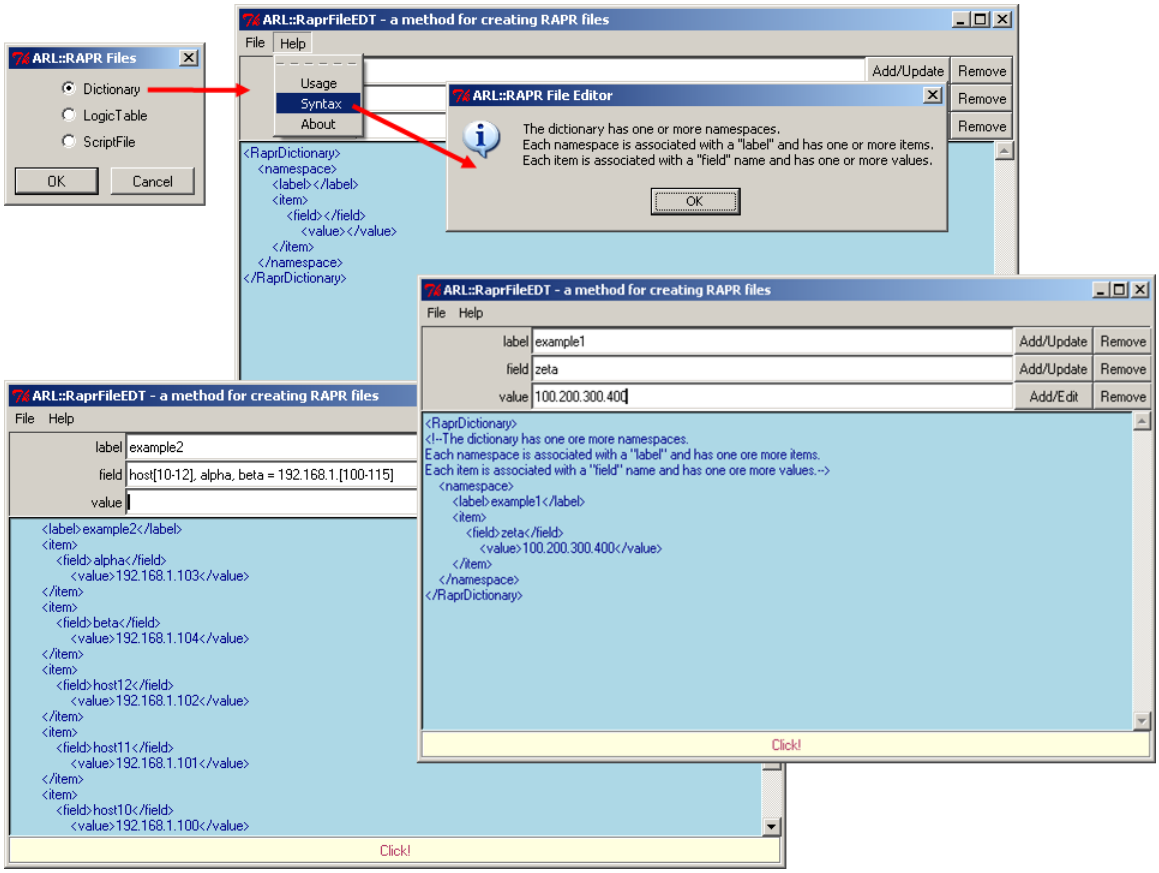

Figure 3. Editing a RAPR dictionary.

2.2 The RAPR Logic-Table Editor

A RAPR logic table describes event-driven behavior of a communication scenario. Each table consists of one or more states. Each state has one or more logicids. The data structure of each logicid has an **id** field, a **percent** field, and one or more **entry** fields. Each entry defines the behavior associated with a given logicid. The **percent** field defines the probability of the triggering event, and it ranges from 0.0 to 1.0.

When a RAPR logic table is to be edited, the ARL RaprEdt tool displays a set of editing options that are specifically designed for editing a logic table. Figure 4 displays the screenshot of the ARL RaprEdt when it deals with a RAPR logic table. The rule for creating a logic table and the XML tags are shown to its users for informational purposes only, i.e., the user does not need to deal with the XML tags. The contents of the field entry box can be manually typed in or left blank. When it is left blank, clicking the **Add/Edit** button will display a window showing the various events and commands from which the user can select. The procedure for completing the process is similar to the process of creating a RAPR script file, which is described in section 2.3, the RAPR Script section.

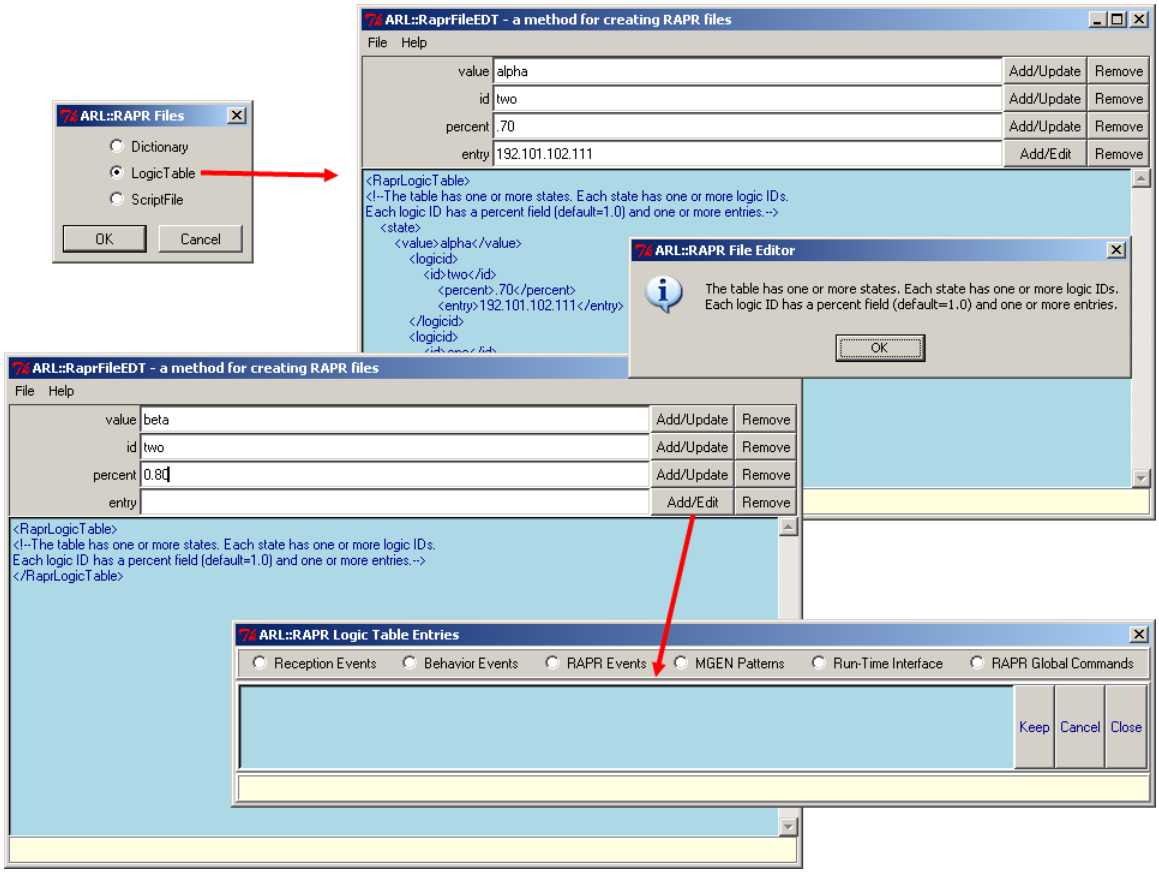

Figure 4. Editing a RAPR logic table.

2.3 The RAPR Script Editor

The RAPR script specifies an event, a Multi-Generator (MGEN) pattern, run-time interface, or RAPR command. An event can be either a reception event, a behavior event, or a RAPR event. The RAPR script editor provides the user two different ways to create a RAPR script: (1) manually typing in a script or (2) using a series of pop-up windows to select an appropriate script and to set the values of its parameters if it requires.

Clicking the **Add/Edit** button while the **Command** field is empty displays a window showing the various events and commands from which the user can select. Figure 5 shows the MGEN Patterns option and its POISSON pattern, among the three patterns, are selected. Once an option is surely chosen, the user has to press the **Keep** button to retain the option. The RAPR script editor tracks a list of selected options that the user wants to keep. Figures 6–8 show various ways for creating a RAPR script by specifying a reception event, a global command, a behavior command, a RAPR event, or a run-time interface directive.

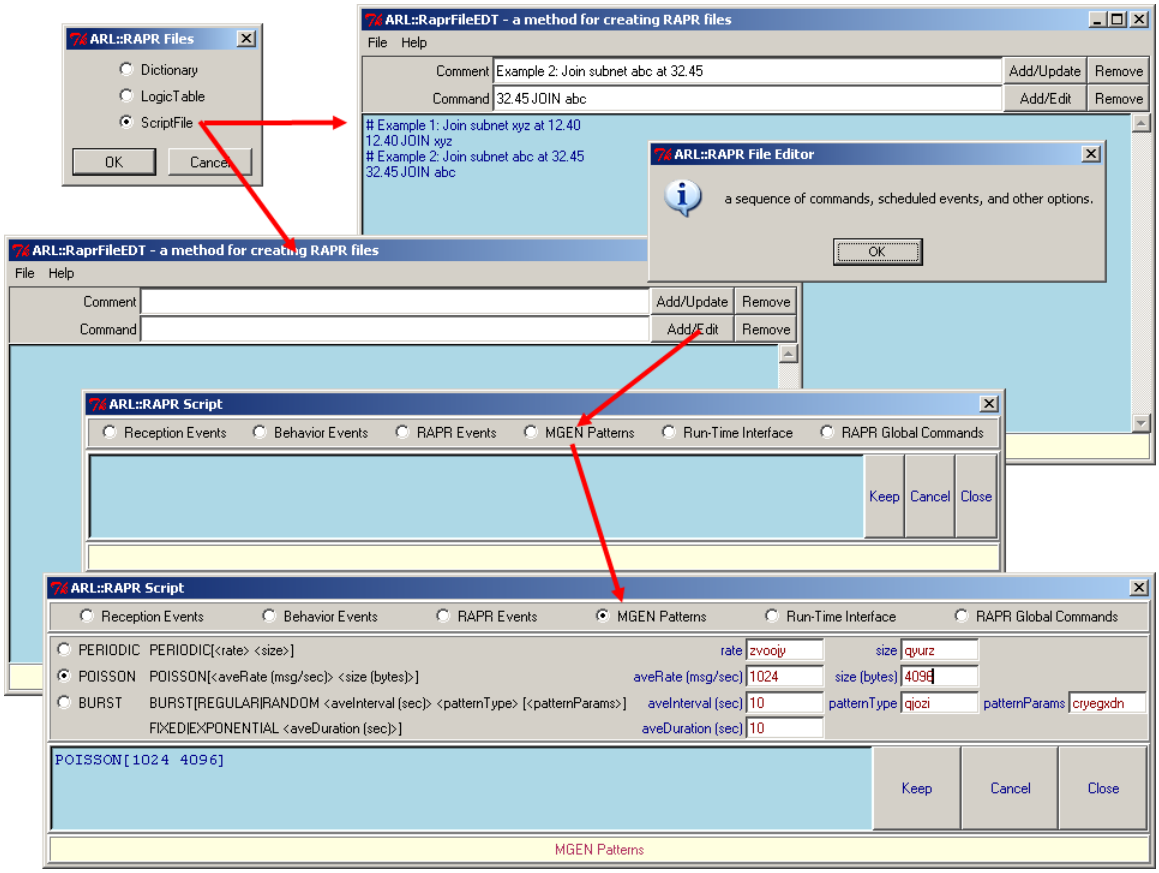

Figure 5. Editing a RAPR script by selecting an MGEN pattern.

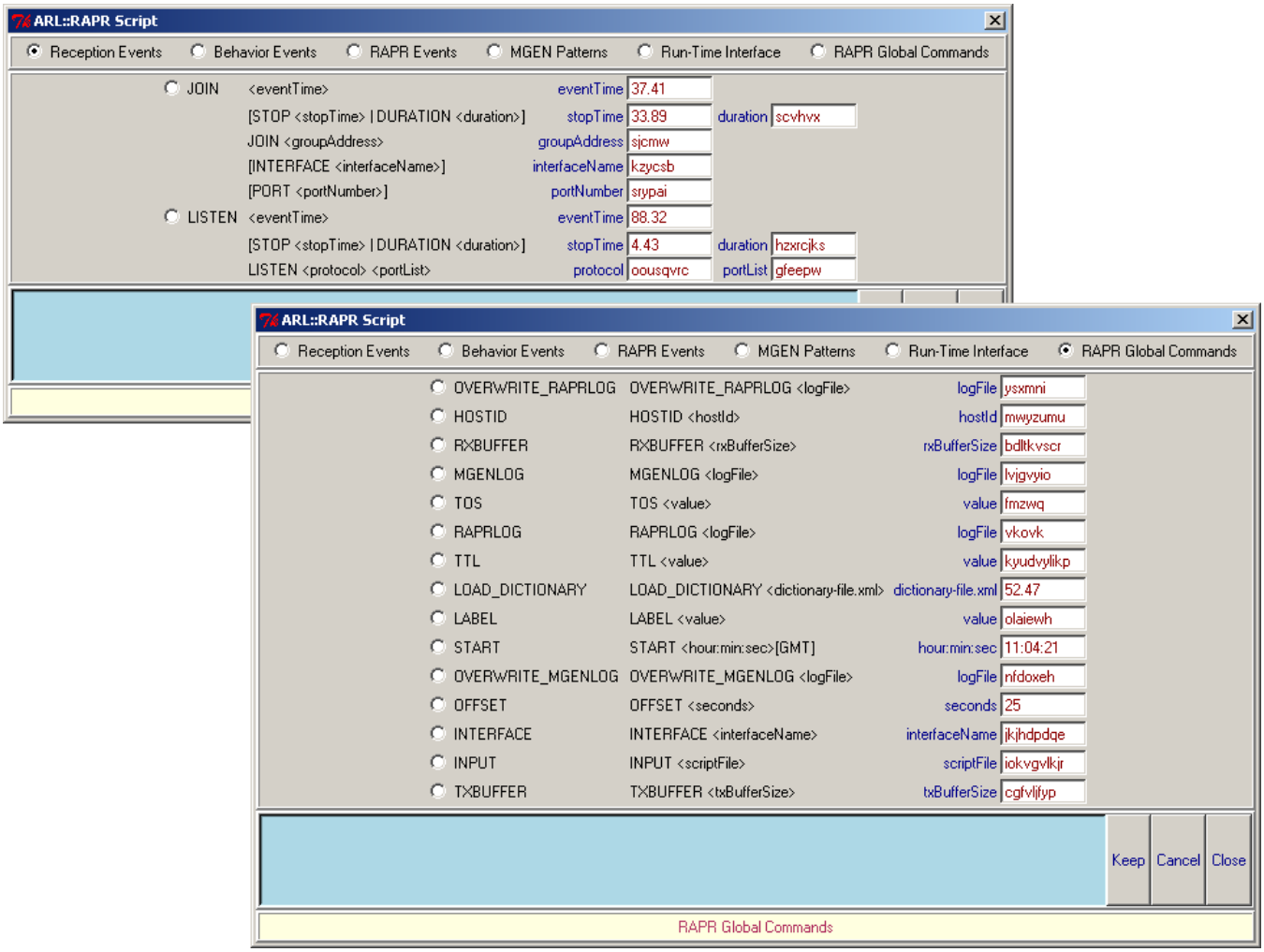

Figure 6. List of reception events and global commands.

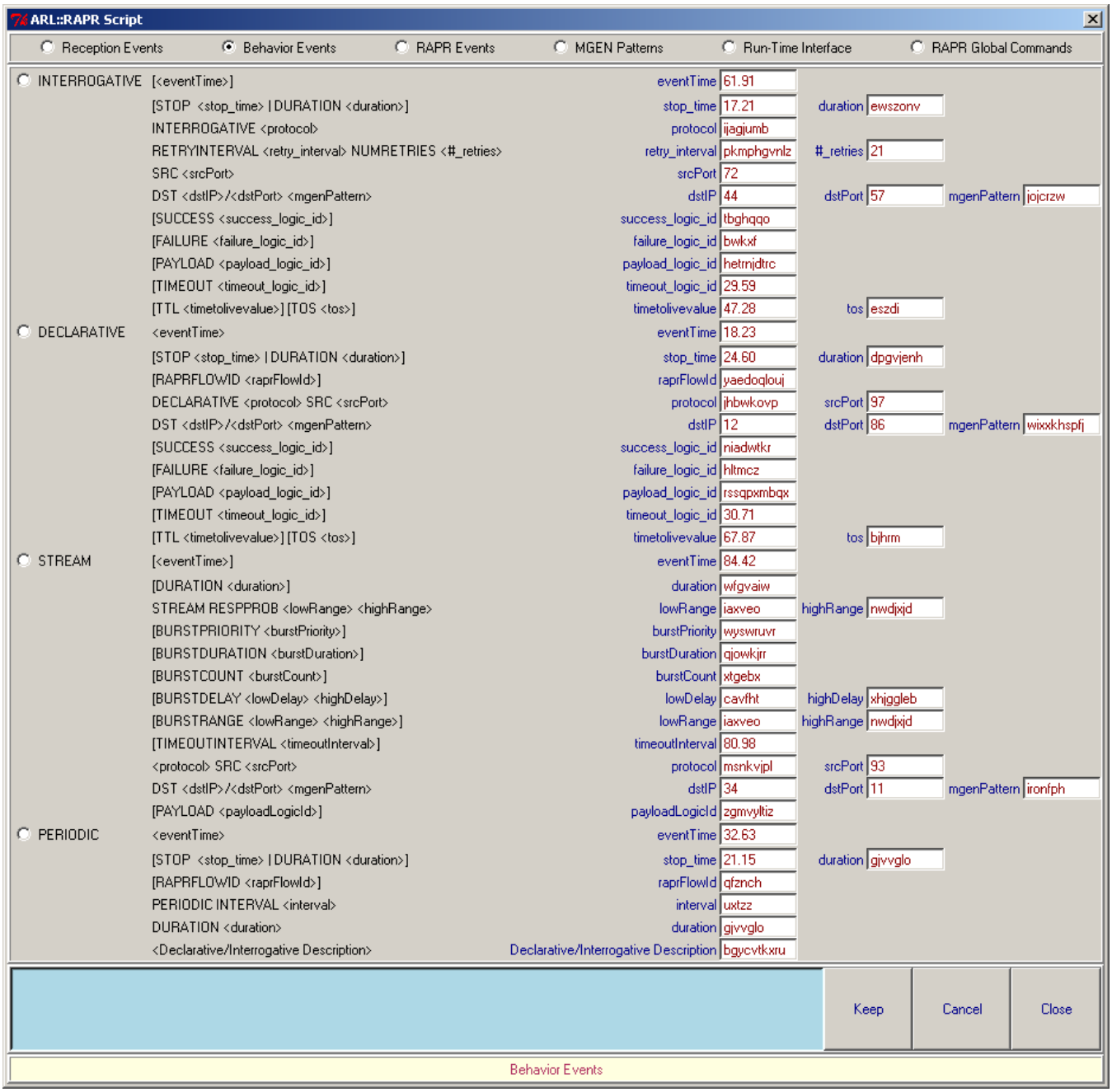

Figure 7. List of RAPR behavior events.

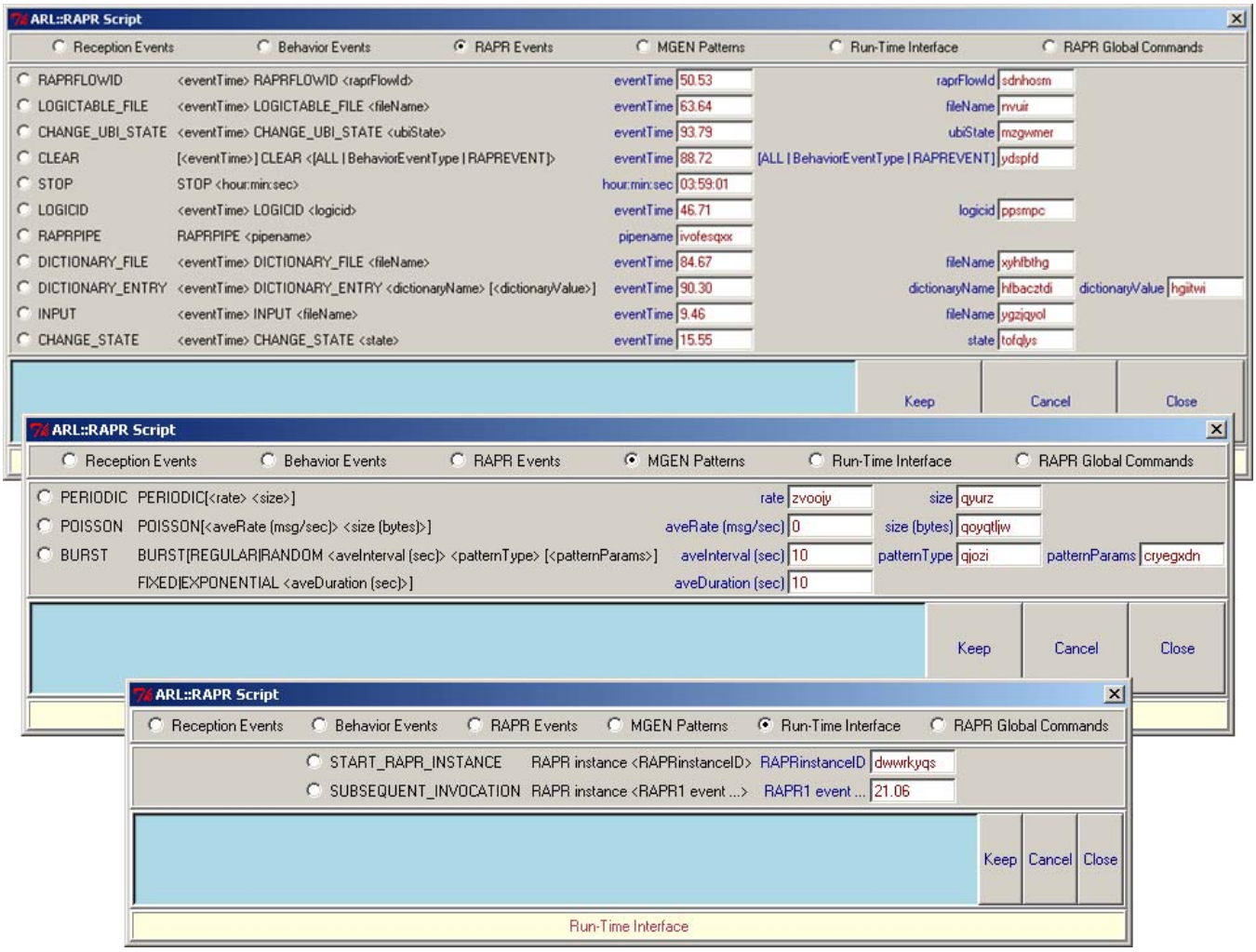

Figure 8. List of RAPR events, MGEN patterns, and run-time interfaces.

3. Conclusion

The successful development of the ARL RaprEdt tool facilitates the creation RAPR files at ARL. Using the tool, a communication engineer can focus on the development of a communication scenario without being concerned with the mechanical structure of the files; especially, the XML files. Once a dictionary and a logic table are created, their contents are embedded in appropriate script files for specifying and running various communication scenarios.

Acronyms

ARL	U.S. Army Research Laboratory
FY08	fiscal year 2008
GUIs	graphical user interfaces
IA	information assurance
MANET	mobile ad hoc network
MGEN	Multi-Generator
NRL	U.S. Naval Research Laboratory
RAPR	Real-time Application Representative
VMs	virtual machines
WEL	wireless emulation laboratory
XML	extensible markup language

No. of Copies	Organization
1 (PDF ONLY)	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP (ELECTRONIC COPY) 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1 HC	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK TP TECHL LIB T LANDFRIED BLDG 4600 APG MD 21005-5066
6 HCs	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK T TECHL PUB ATTN AMSRD ARL CI OK TL TECHL LIB ATTN IMNE ALC IMS MAIL & RECORDS MGMT AMSRL ARL CI NT N IVANIC R HARDY B NGUYEN ADELPHI MD 20783-1197
Total:	8 (1 PDF, 7 HCs)